
# DNN-Dataflow-Hardware Co-Design for Enabling Pervasive General-Purpose AI

http://synergy.ece.gatech.edu

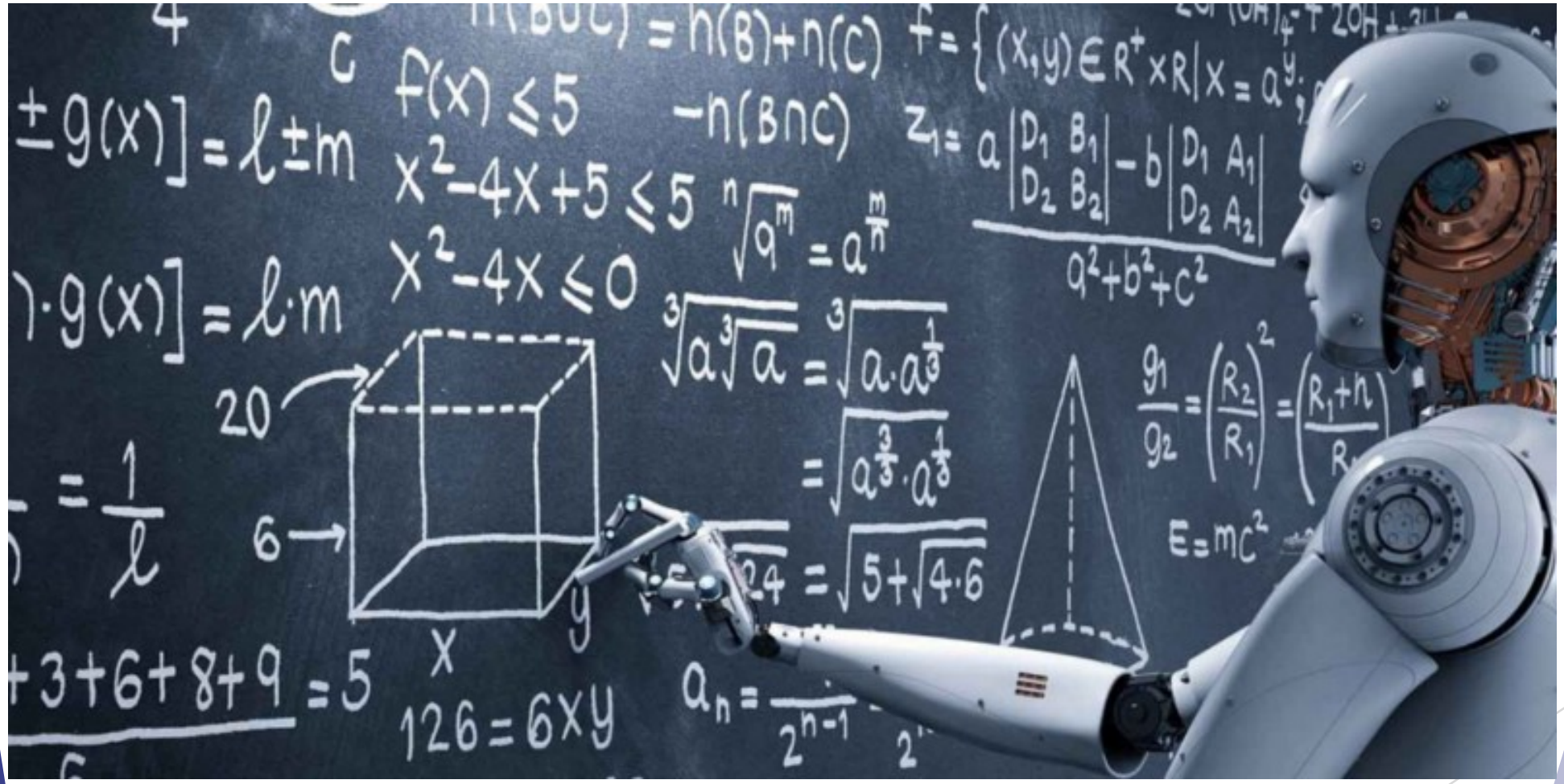
**Tushar Krishna**

Assistant Professor
School of ECE
Georgia Institute of Technology
tushar@ece.gatech.edu

MAESTRO
MAERI
GeneSys

CRNCH Summit
November 2, 2018

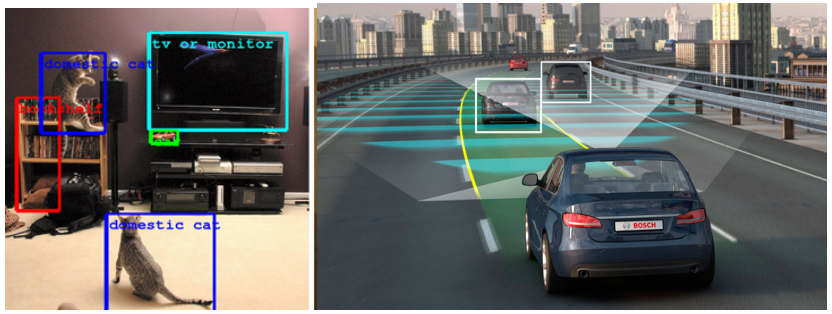
# The Dream!



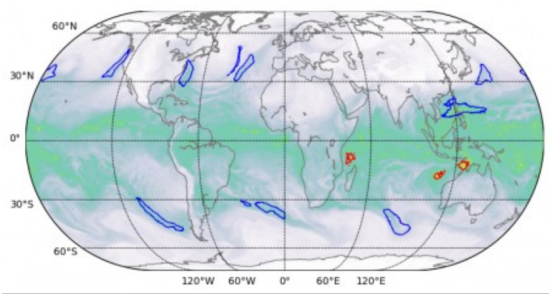
# Deep Learning Applications

"AI is the new electricity" – Andrew Ng

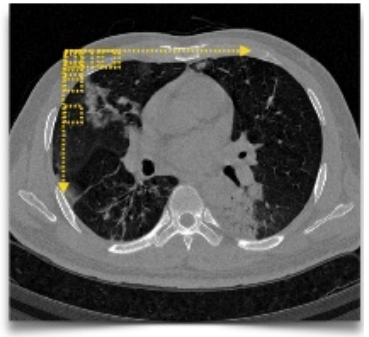
### Object Detection




### Image Segmentation

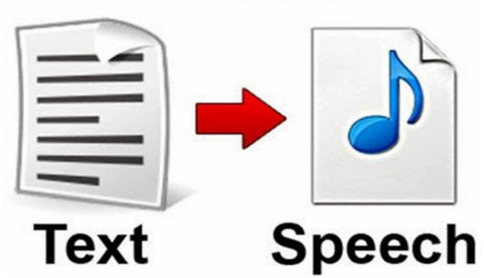



### Medical Imaging



### Speech Recognition

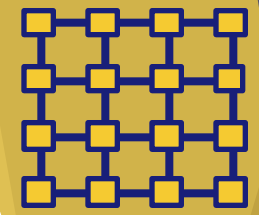

### Text to Speech




### Recommendations

### Games

# What is a Deep Neural Network?

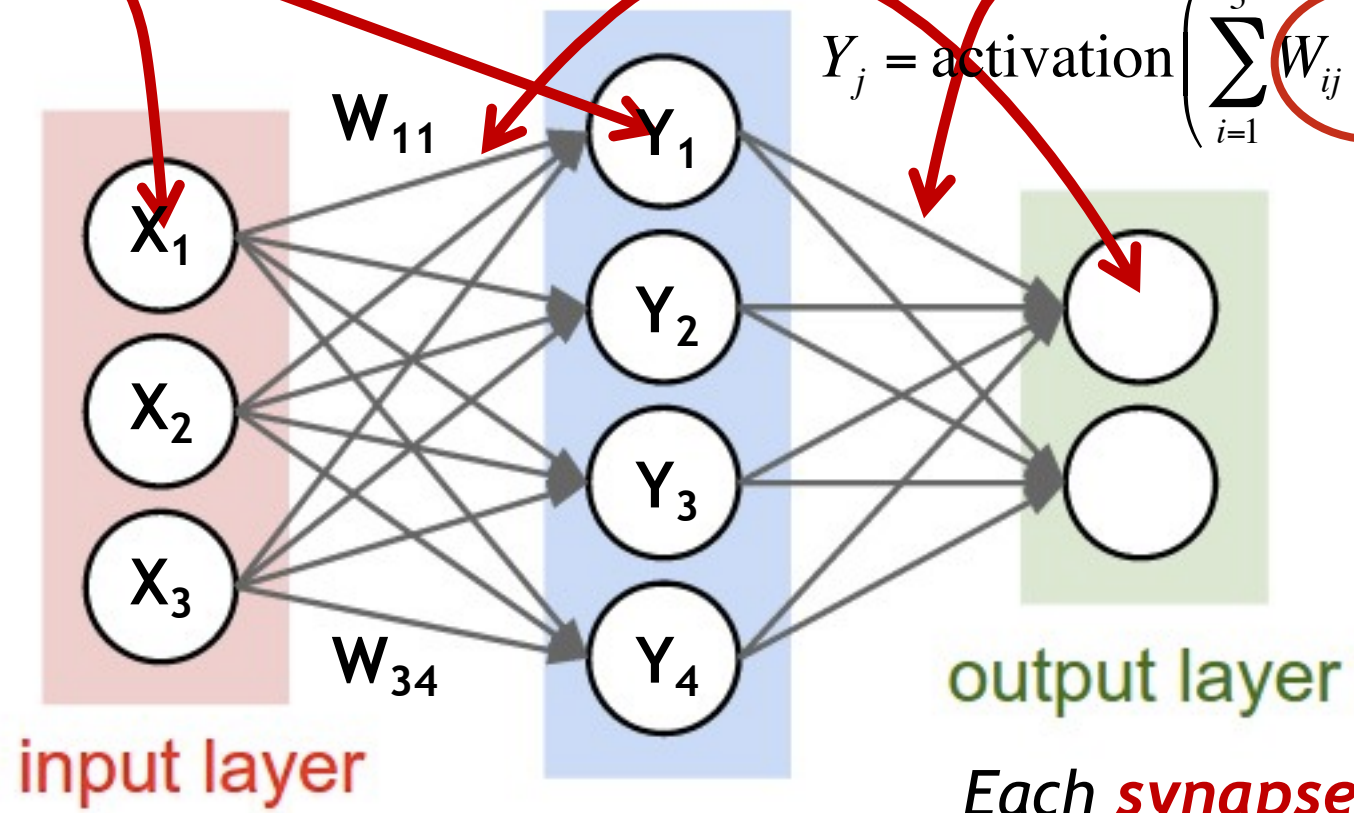


**Neurons**

**Synapses**

$$Y_j = \text{activation}\left(\sum_{i=1}^{3} W_{ij} \times X_i\right)$$

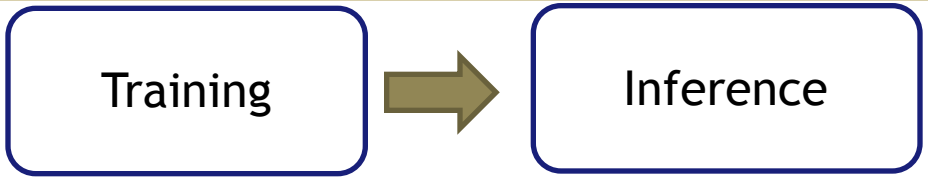
Weighted Sum



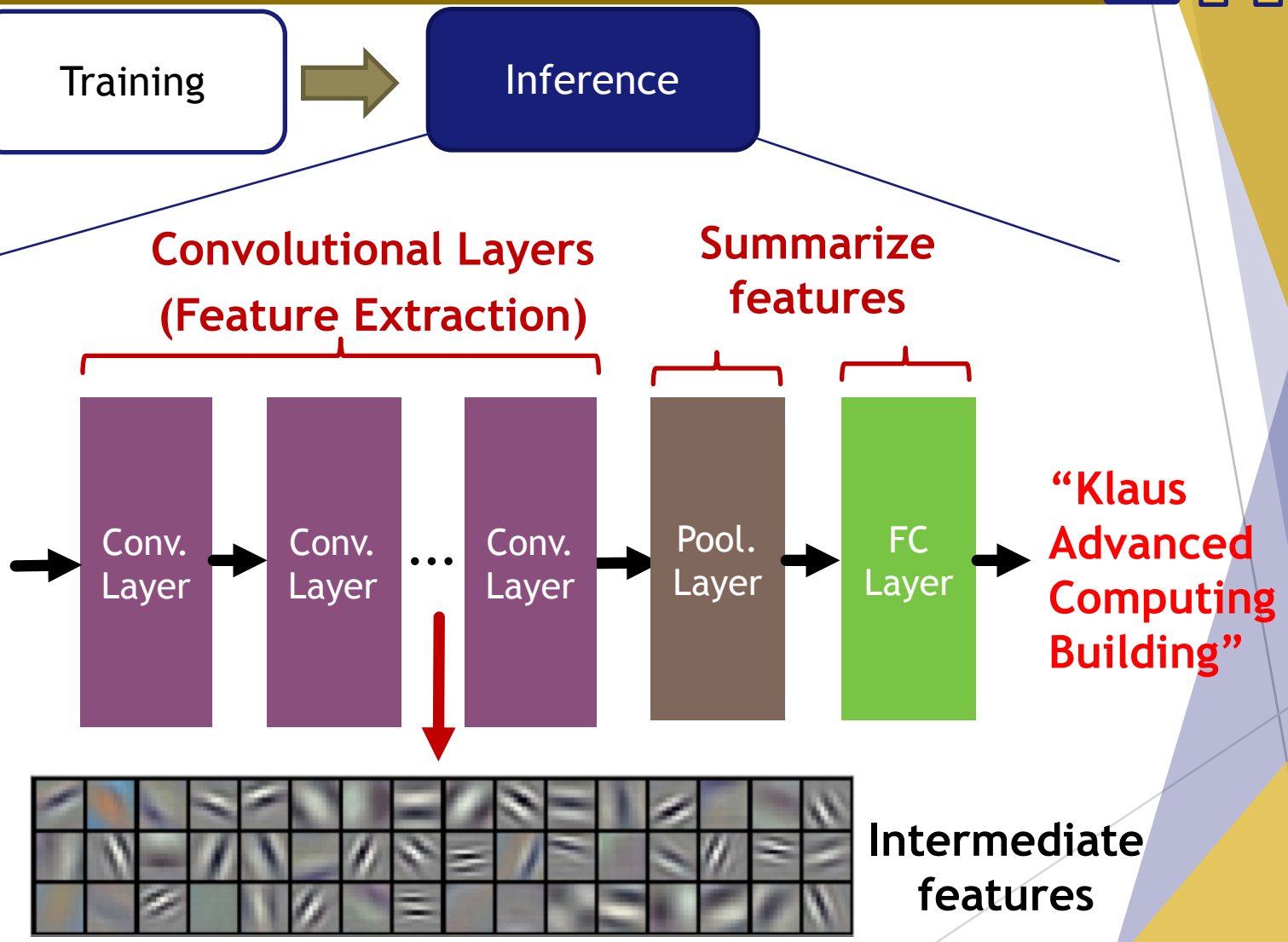

[Image Source: Stanford]

*Each **synapse** has a **weight** for neuron **activation***

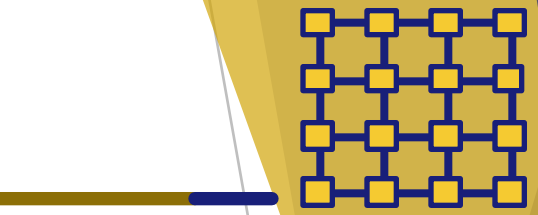
# Deep Learning Landscape



# Deep Learning Landscape



# Deep Learning Landscape



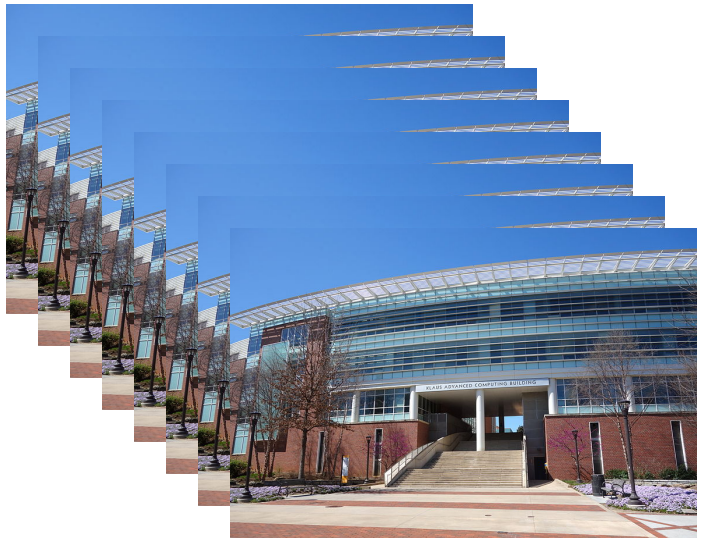
Training

Inference

*Error*

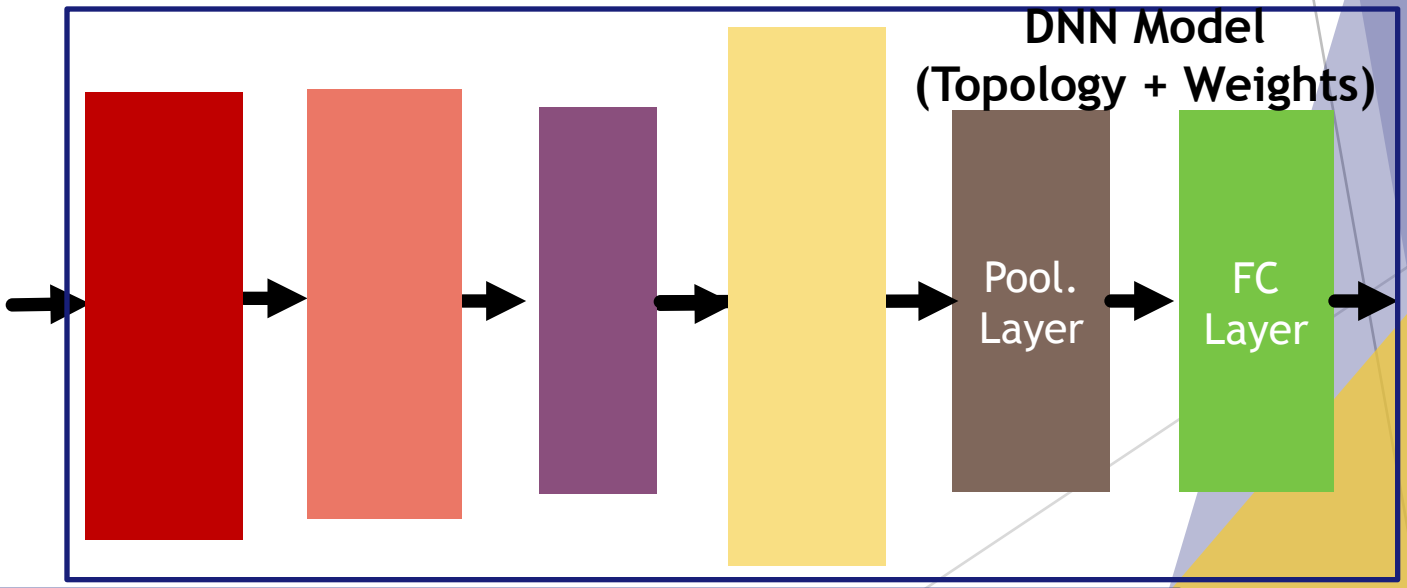
ML Practitioner

**Labelled Datasets**



**"Klaus Advanced Computing Building"**

**DNN Model (Topology + Weights)**



# Computation Platforms

Training → Inference

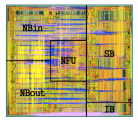
**HPC cluster**

ARM Trillum

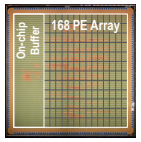
NVDLA

Apple Neural Engine

CambriconX



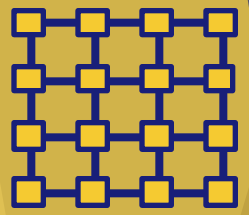

ShiDianNao



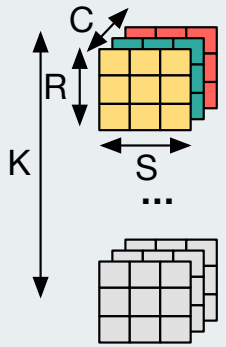

Eyeriss

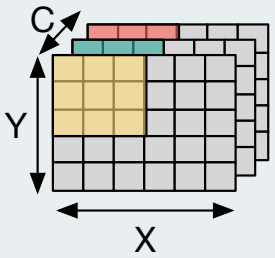
**Accelerators**

# Challenges in Designing and Deploying AI

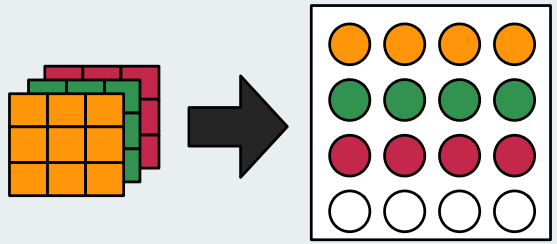


Training → Inference

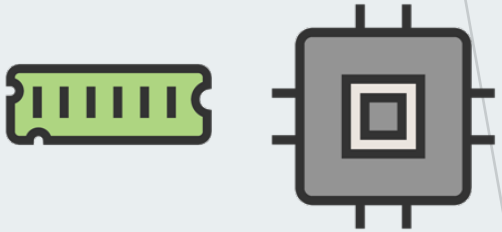



DNN Architecture



Mapping (Dataflow)

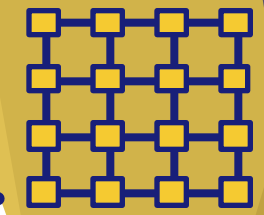


Accelerator Microarchitecture

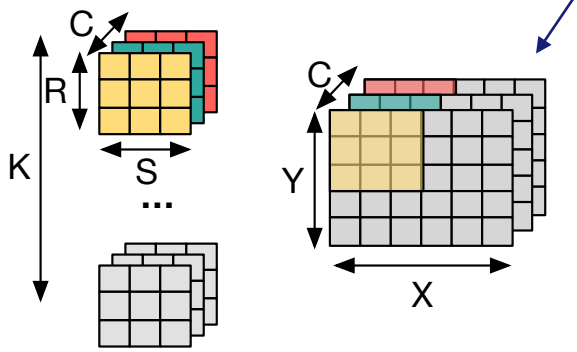
Energy

Runtime

# Outline of Talk

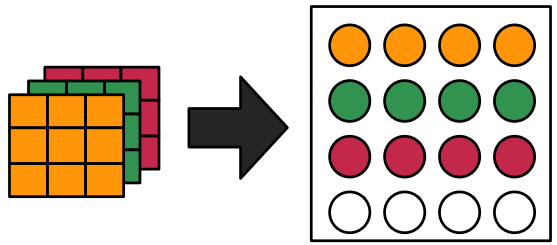


Training → Inference



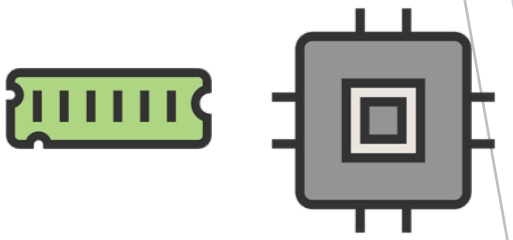

DNN Architecture

GeneSys



Mapping (Dataflow)

MAESTRO



Accelerator Microarchitecture

MAERI

# Deep Learning Landscape

What if

Training

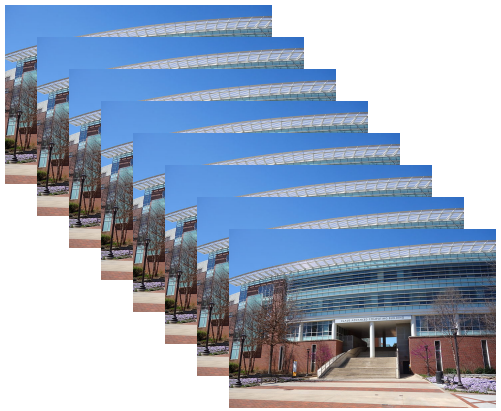
Inference

HPC Cluster

ML Practitioner

Error

Labelled Datasets



"Klaus Advanced Computing Building"

??

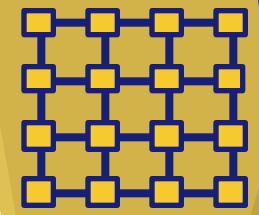
DNN Model (Topology + Weights)

Pool. Layer

FC Layer

Deep Learning not viable for continuous learning

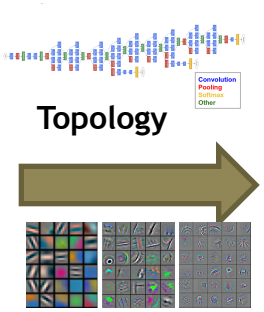
# Template for Continuous Learning



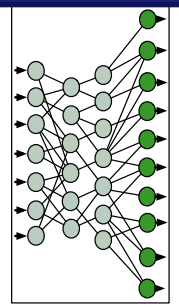
**Robust**

??

**Learning Agent**




**Continuous**

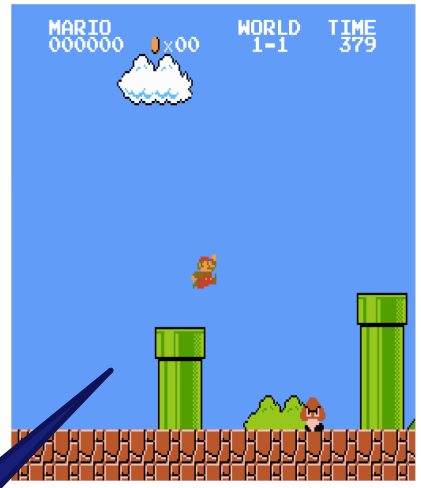


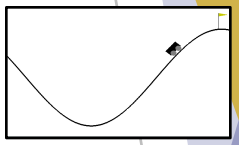
**Inference Agent**

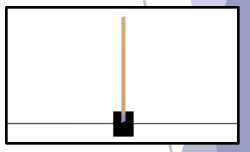
Action

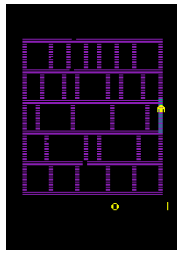
Reward

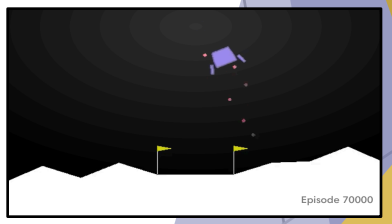
**Environment**

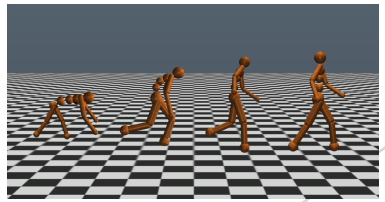






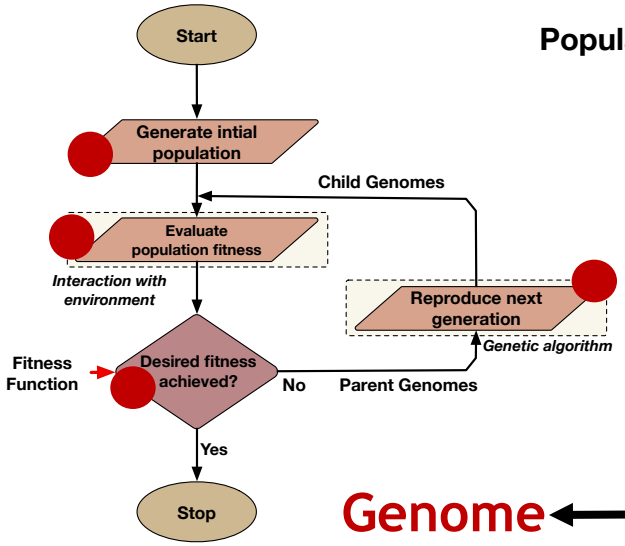



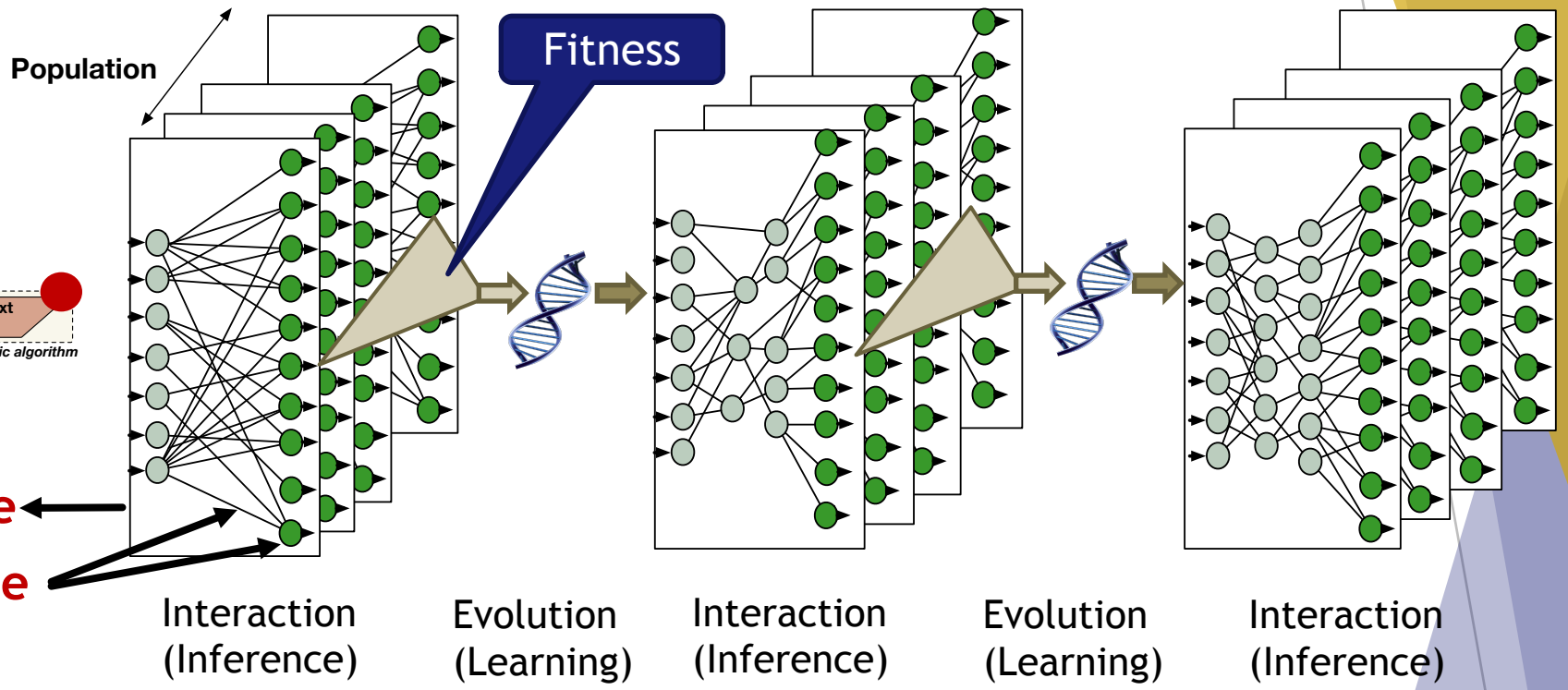
Accumulated Rewards

Learn how to improve at one task

Learn multiple tasks

# Neuro-Evolutionary (NE) Algorithm



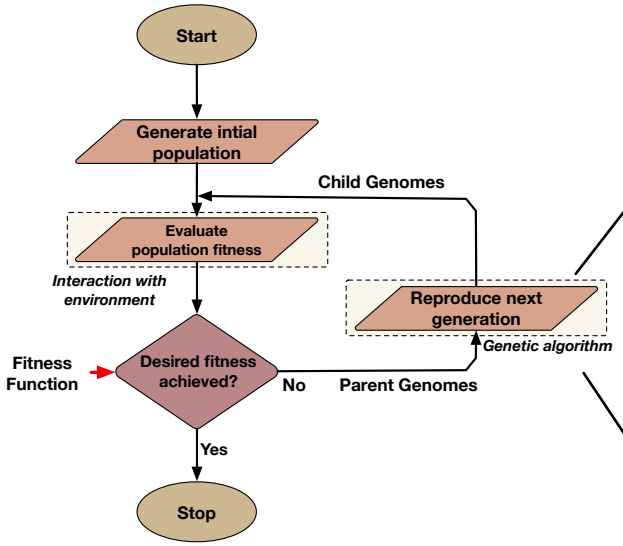

Neural Network (NN) expressed as a graph

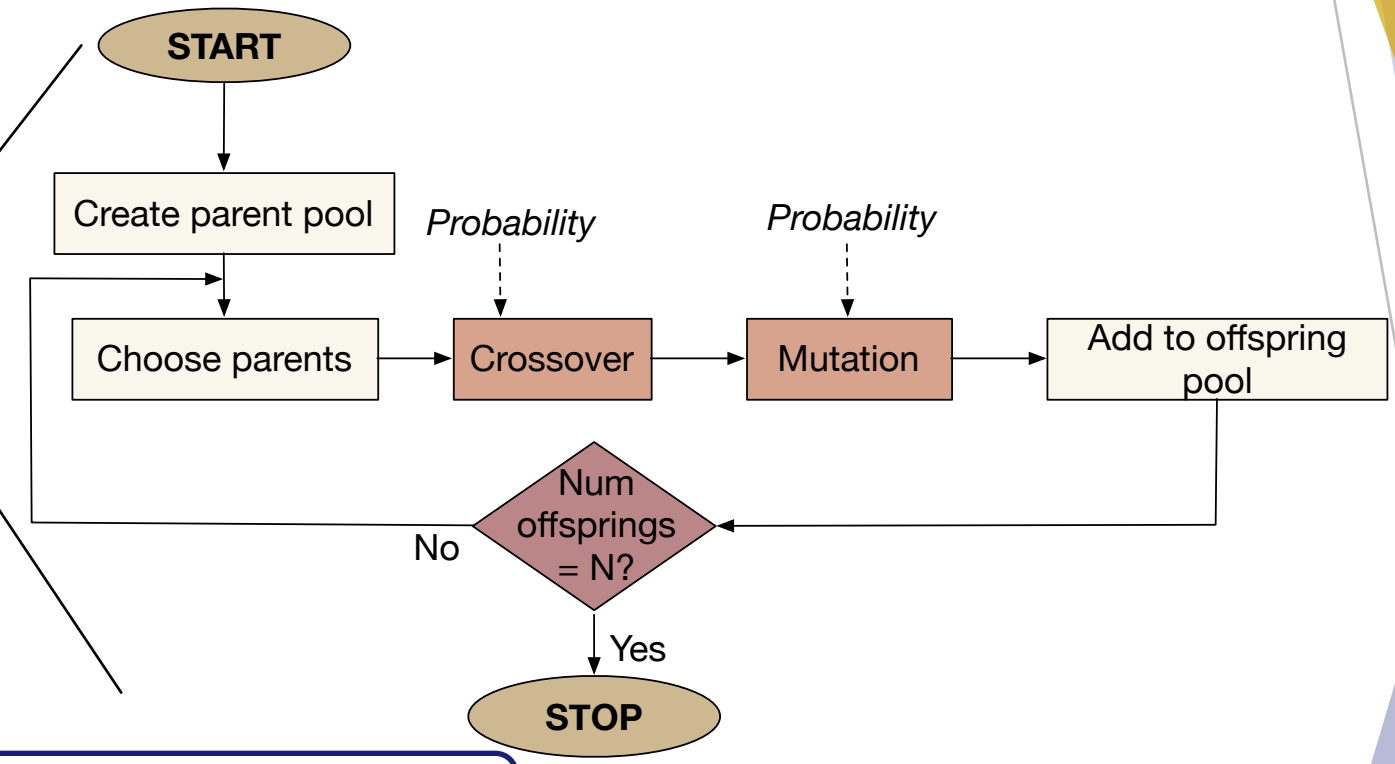
**Gene**: Vertex or Edge in the graph

**Genome**: Collection of all genes (i.e., a NN)

[1] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, *10*(2), 99-127.

# Neuro-Evolutionary (NE) Algorithm




Neural Network (NN) expressed as a graph

**Gene**: Vertex or Edge in the graph

**Genome**: Collection of all genes (i.e., a NN)

**NeuroEvolution of Augmented Topologies (NEAT) [1]**

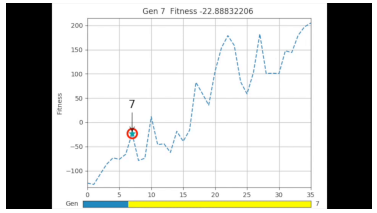

[1] Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, *10*(2), 99-127.

# Properties of NE algorithms

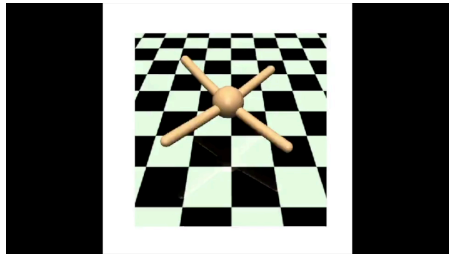
## Algorithmic

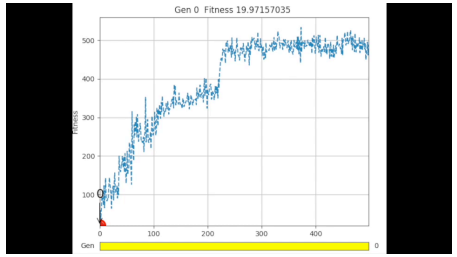
Robustness

No Training




Change fitness function






## Systems

Massive Parallelism

Genomes within Population

Genes within a Genome

Low Memory Footprint

Only store genomes in current generation

No backprop

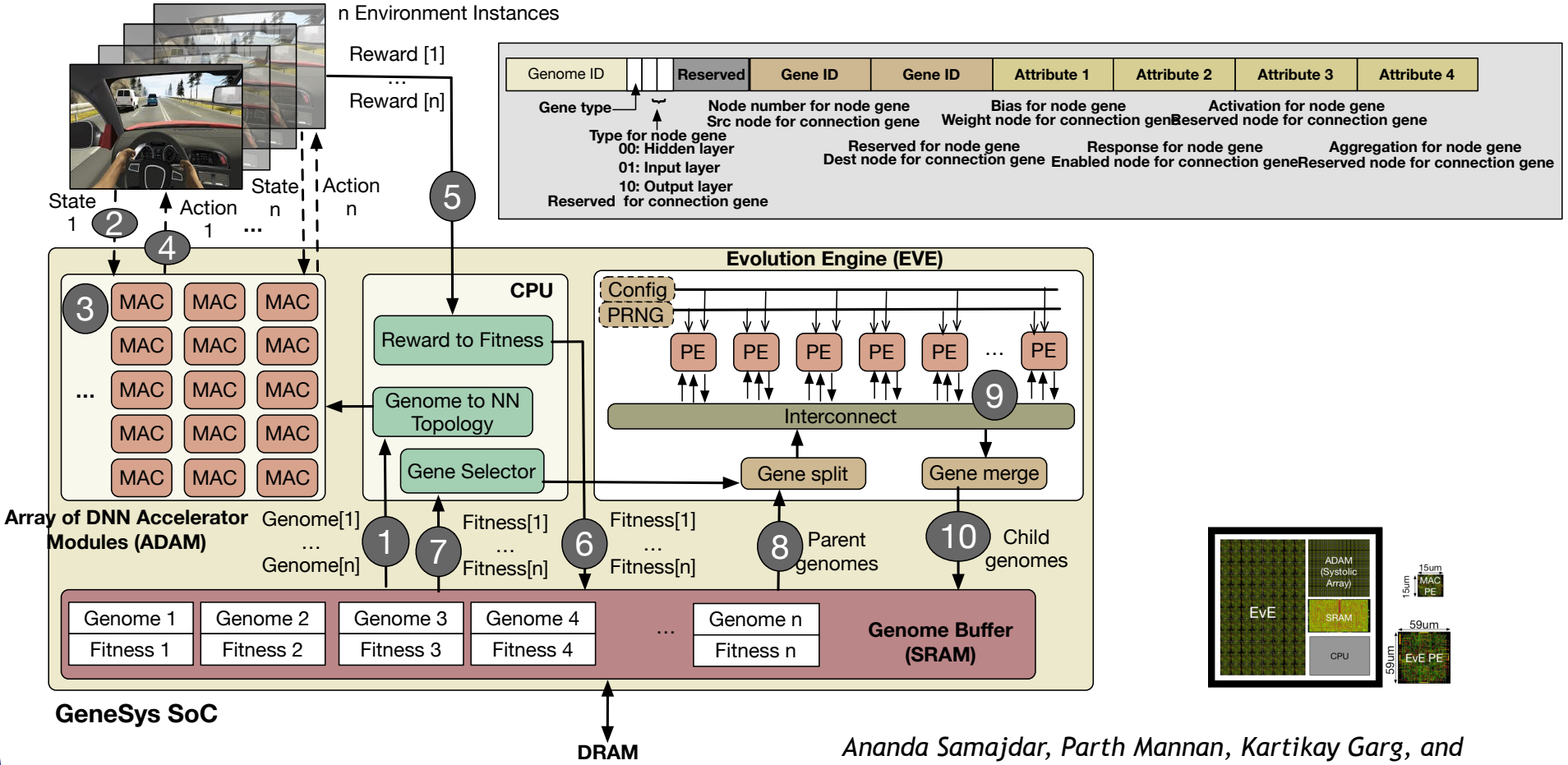
No gradiant calculations or storage

Simple HW-friendly Ops

MACs in Inference
Crossover and Mutation in Evolution

**NE is viable for continuous learning**

# GeneSys SoC

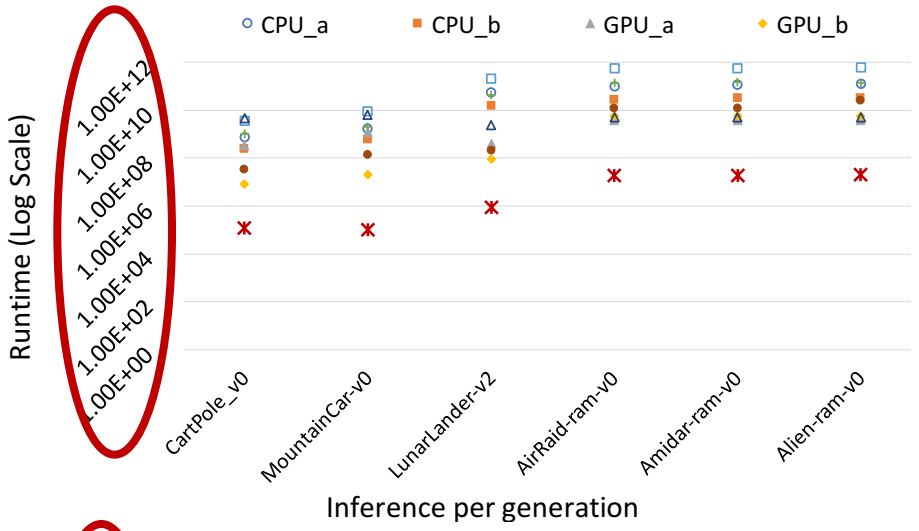


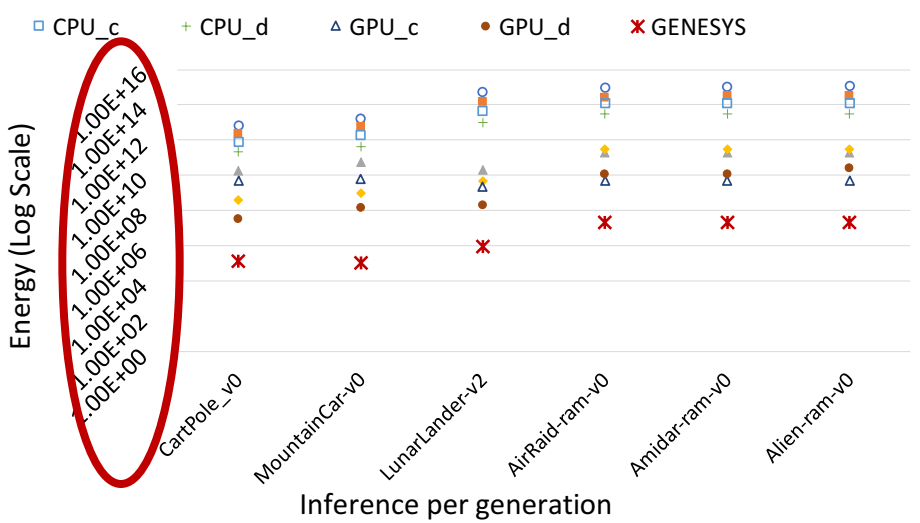


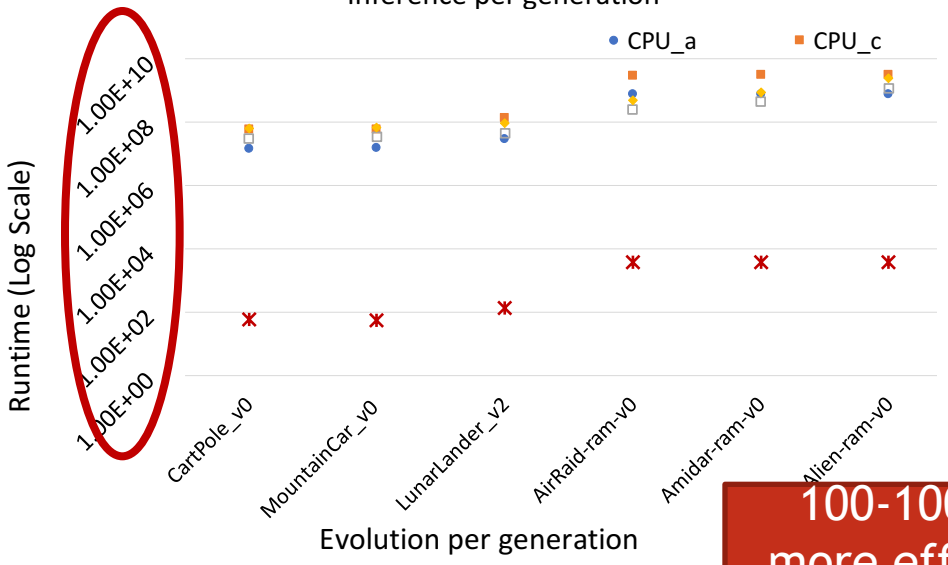


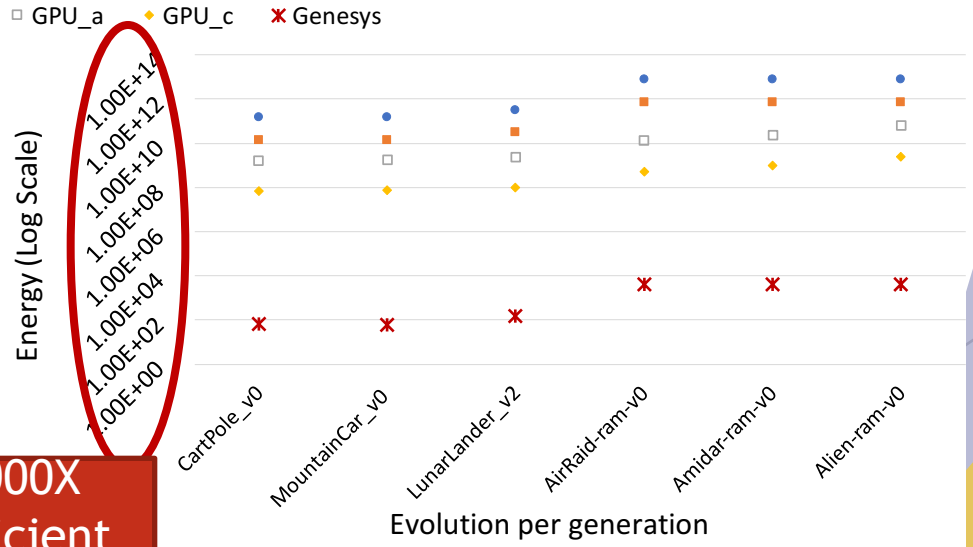
*Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna*
***GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware***
*MICRO 2018*

# GeneSys ASIC: Runtime and Energy



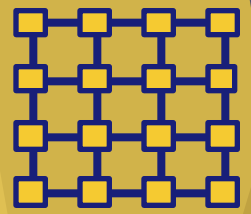



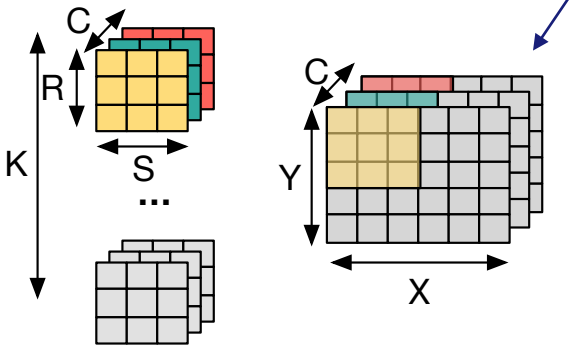




100-10000X more efficient

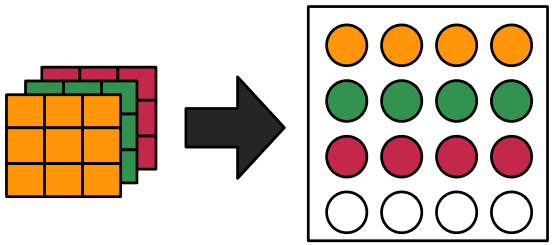
# Outline of Talk



Training → Inference

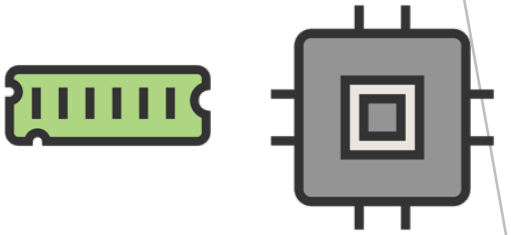



DNN Architecture



Mapping (Dataflow)

MAESTRO



Accelerator Microarchitecture

MAERI

GeneSys

# Why do we need DNN accelerators?

- **Millions of Parameters (i.e., weights)**
  - Billions of computations → **Need lots of parallel compute**

| DNN Topology | Number of Weights |
|---|---|
| AlexNet (2012) | 3.98M |
| VGGnet-16 (2014) | 28.25M |
| GoogleNet (2015) | 6.77M |
| Resnet-20 (2016) | 0.27M |
| Resnet-110 (2016) | 1.7M |

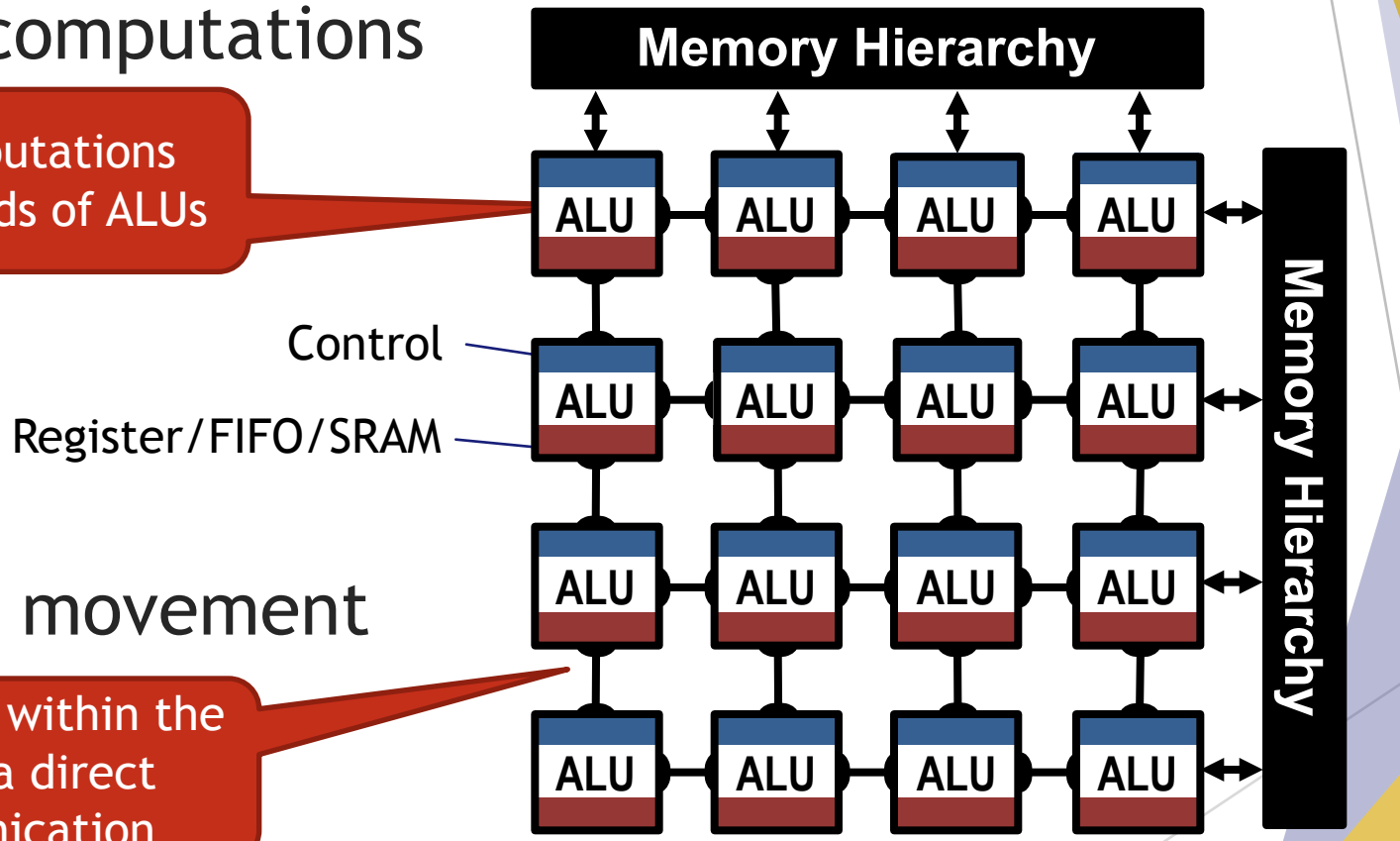
This makes CPUs inefficient

  - Heavy data movement → **Need to reduce energy**


This makes GPUs inefficient

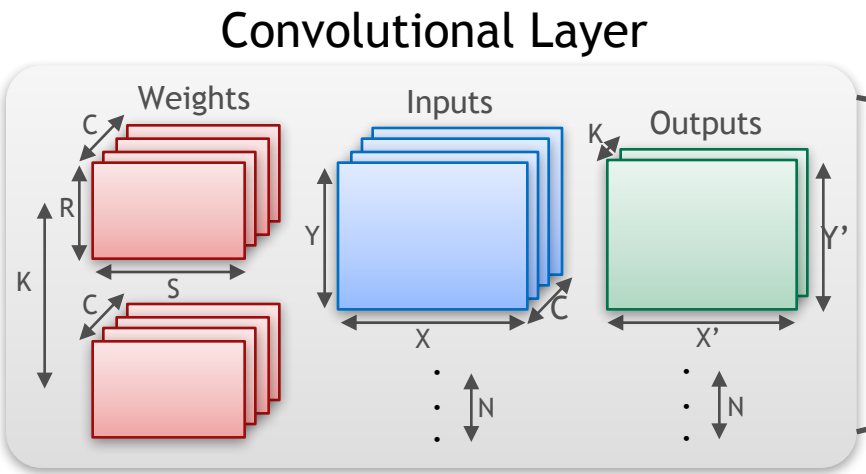
# Spatial (or Dataflow) Accelerators

- **Millions of Parameters (i.e., weights)**
  - Billions of computations
    - Spread computations across hundreds of ALUs
  - Heavy data movement
    - Reuse data within the array via direct communication

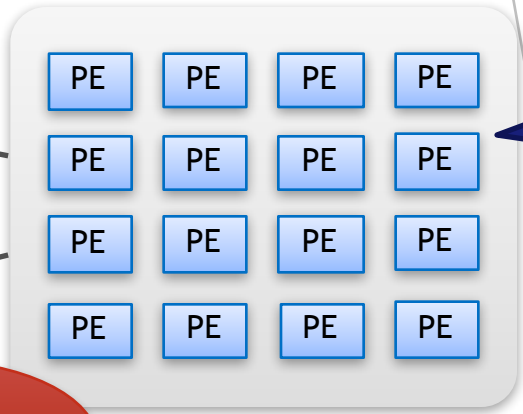



*Examples: MIT Eyeriss, Google TPU, …*

# High-Dimensional Compute → HW

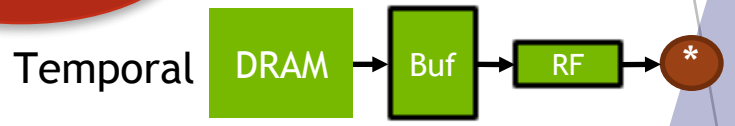


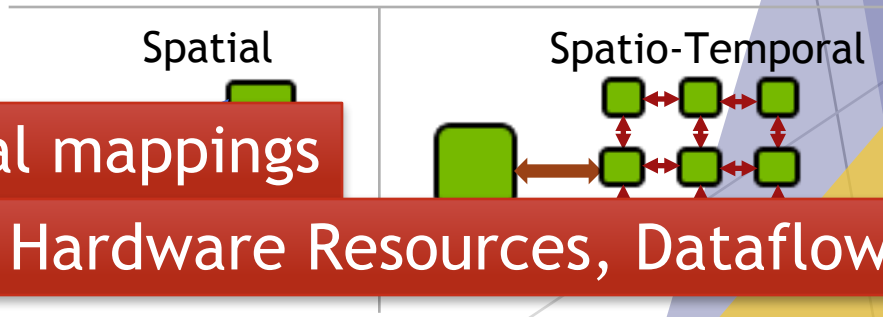

map




Accelerator HW (ASIC or FPGA or HPC System)

Data Reuse




- **7D Computation Space**
  - R * S * X * Y * C * K * N
- **4D Operand / Result Spaces -**
  - Weights - R * S * C * K
  - Inputs - X * Y * C * N
  - Outputs - X' * Y' * K

Millions of non-trivial mappings

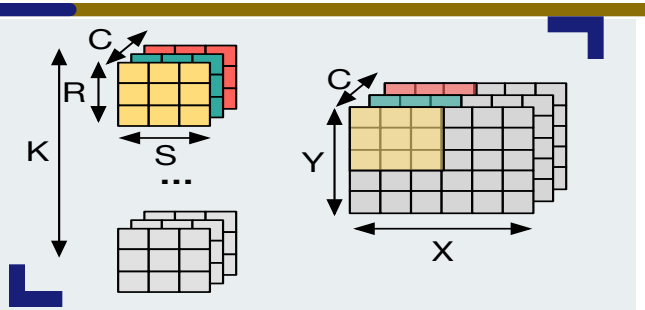
Energy Benefits = f(Dimension Sizes, Hardware Resources, Dataflow)

How do we explore all possible dataflows?

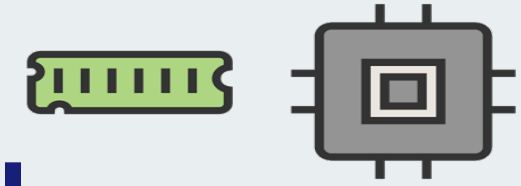


*Slide Adapted From: Joel Emer, Angshuman Parashar (NVIDIA)*

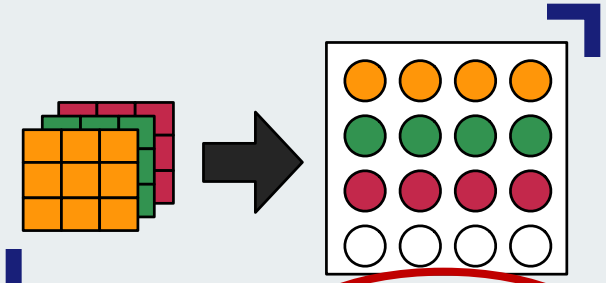
# MAESTRO: Analytical Cost/Benefit Model




DNN Layer Sizes



HW Resources



Mapping (Dataflow)

MAESTRO

- Abstract HW Model
- Data Reuse Analysis
- Communication Analysis
- Computation Analysis

## Buffer Analysis

- Size Requirement
- Access Count (Energy)

## NoC Analysis

- BW Requirement
- NoC Activity Count

## Runtime Analysis

- Roofline Throughput
- Expected Runtime



*H. Kwon et. al., "An Analytic Model for Cost-Benefit Analysis of Dataflows in DNN Accelerators," https://arxiv.org/abs/1805.02566

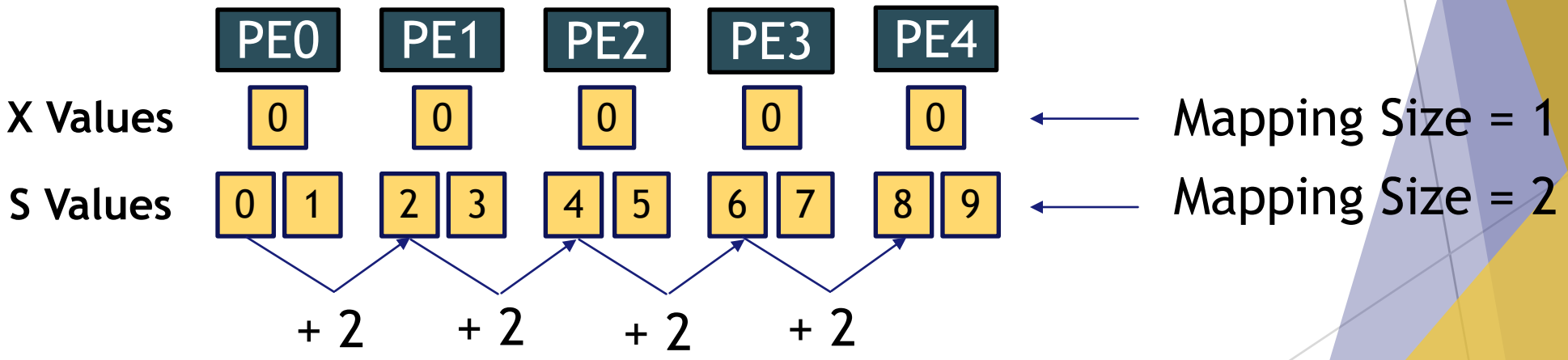
# Describing Dataflows in MAESTRO

```
for(int x = 0; x < 20; x++)
  for(int s = 0; s < 10; s++)
    Output[x] += Weight[s] * Input[x+s]
```

*Parameters:*
*(Mapping size, Offset)*

**MAESTRO: Temporal_Map(1, 1) *X* -> Spatial_Map(2, 2) *S***

Time step = 0

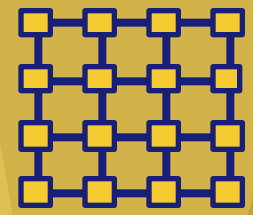



* Temporal_Map: to map the same loop variable set to each PE
* Spatial_Map: to map different loop variable sets to each PE

Replication

Block-Cyclic Distribution

# Input: MAESTRO DSL



```
1  | //Hardware Resource Description
2  | L1Size 64
3  | L2 Size 1024
4  | NoCBW 64
5  | Multcast True
6  | NumPEs 256
7  | ...
8  | //DNN Layer Description
9  | Layer CONV VGG16_C1
10 |  K=64;C=3;R=3;S=3;Y=224;X=224
11 | endLayer
12 |
```

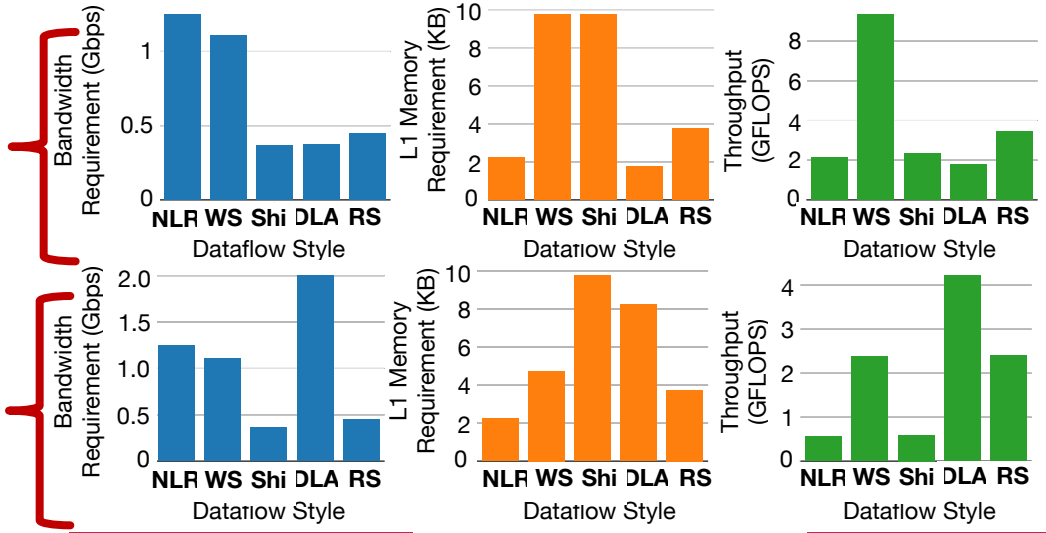
```
13 |  //Mapping (Dataflow) Description
14 | Temporal_Map (1,1) K
15 | -> Temporal_Map (1,1) C
16 | -> Temporal_Map (3,1) Y
17 | -> Tile (3) Y
18 | -> Spatial_Map (1,1) X
19 | -> Unroll R
20 | -> Unroll S
```

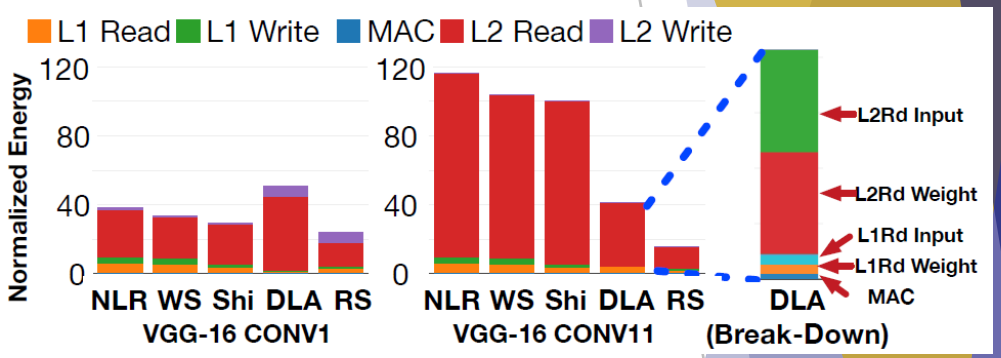
Temporally/Spatially maps iteration variables to each PE

# Output: Cost/Benefit Analysis

Early Layer (C1)

Late Layer (C11)




**NoC BW Analysis**

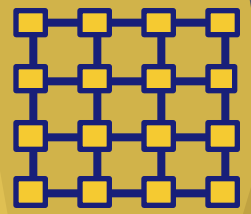
**Buffer Sz Analysis**

**Throughput Analysis**

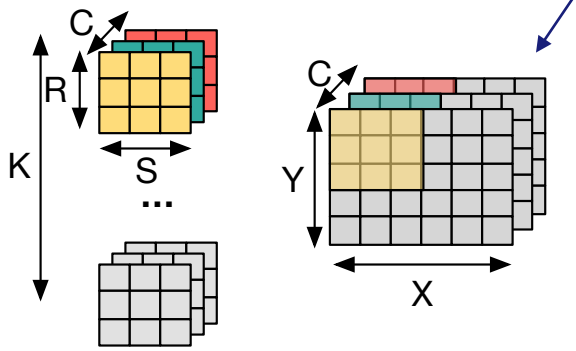
**Energy Analysis**

**No single dataflow is good for every layer**

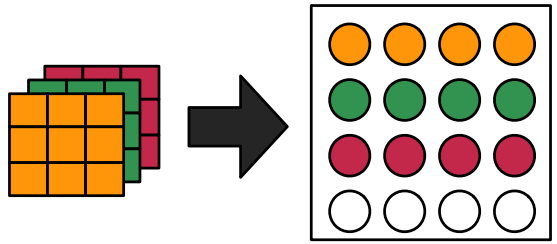
# Outline of Talk



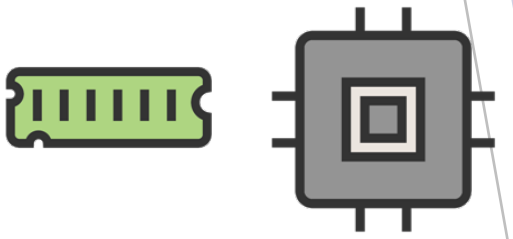
Training → Inference




DNN Architecture



Mapping (Dataflow)



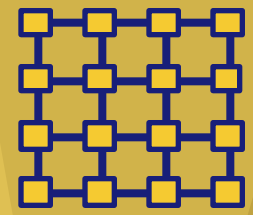
Accelerator Microarchitecture

GeneSys

MAESTRO

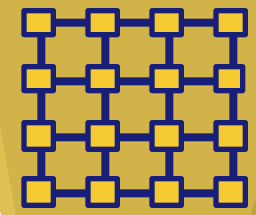
MAERI

# Myriad Dataflows in DNN Accelerators



- **DNN Topologies**
  - Layer size / shape
  - Layer types: Convolution / Pool / FC / LSTM
  - New sub-structure: e.g., Inception in Googlenet
- **Irregular Networks**
  - Weight Pruning during Training
  - Generated by GeneSys
- **Compiler/Mapper Optimization (i.e., MAESTRO)**
  - Loop reordering
  - Loop tiling size
  - Cross-layer mapping

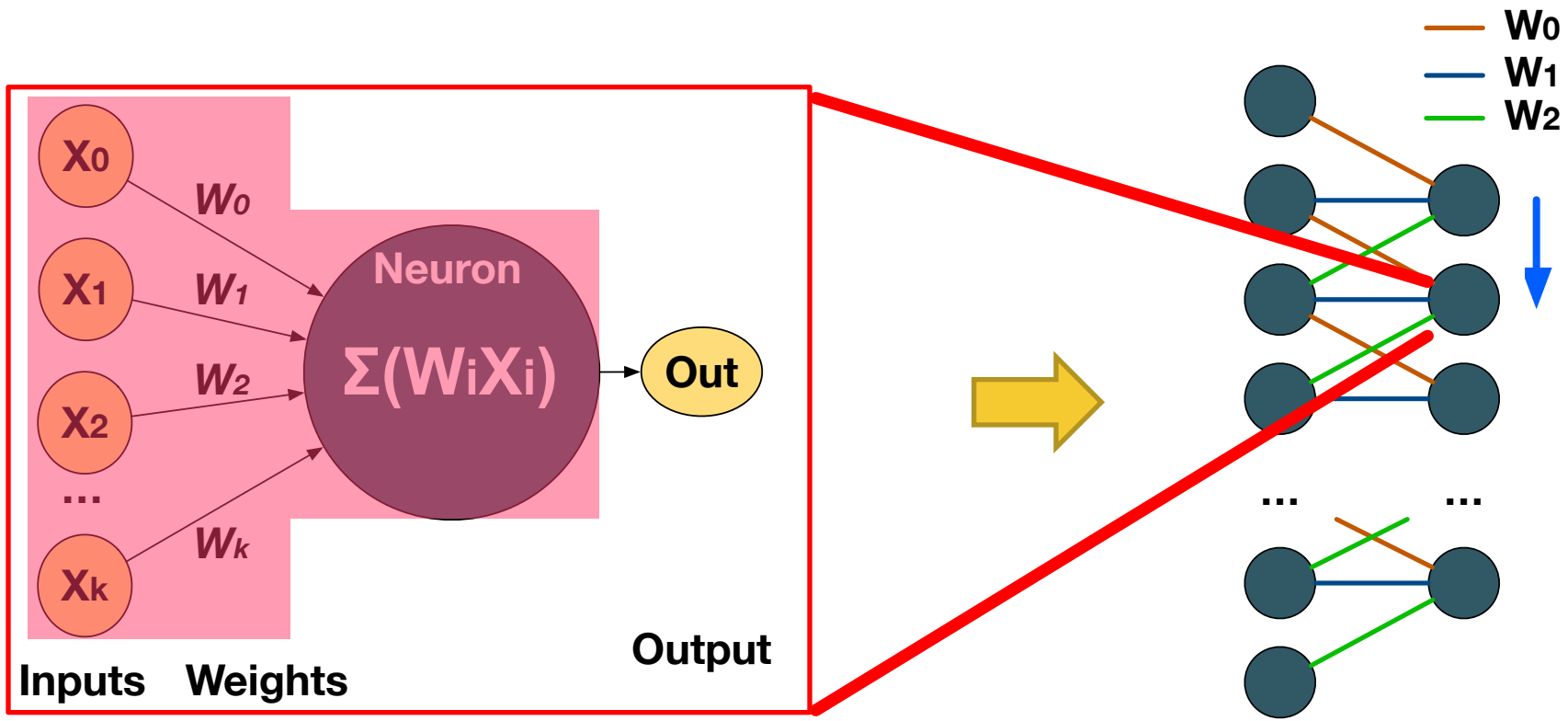
# The current trend for supporting multiple dataflows



- New Dataflow → New Accelerator
  - **Data reuse:** FlexFlow (2017), Eyeriss (2016), ...
  - **Cross-layer:** Fused CNN (2016)
  - **Sparse CNN:** SCNN (2017), EIE(2016), ...
  - **LSTM:** ESE (2017), ...

Can we have one architectural solution that can handle arbitrary dataflows and provides ~100% utilization?

# What is the computation in a DNN?







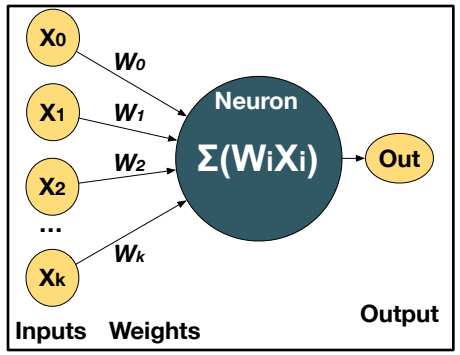
Compute weighted sum

Independent multiplication

Accumulation of partial products

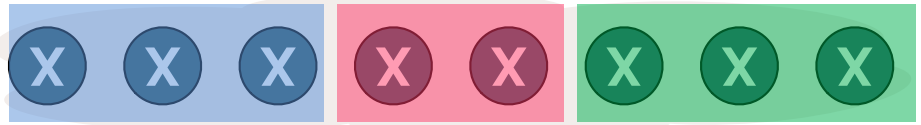
Our Key insight: Each dataflow translates into neurons of different sizes

# The MAERI Abstraction




**Multiplier Pool**

"MultAlloc(2); AddAlloc(1)"



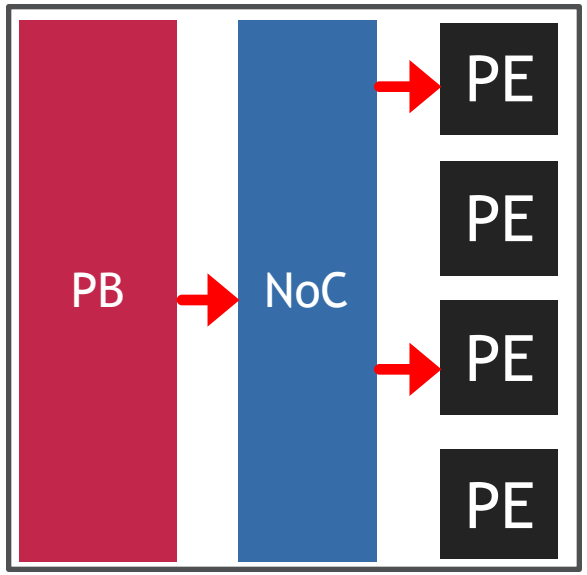

**VN0** **VN1** **VN2**


**Adder Pool**

**Virtual Neuron (VN):** Temporary grouping of compute units for an output

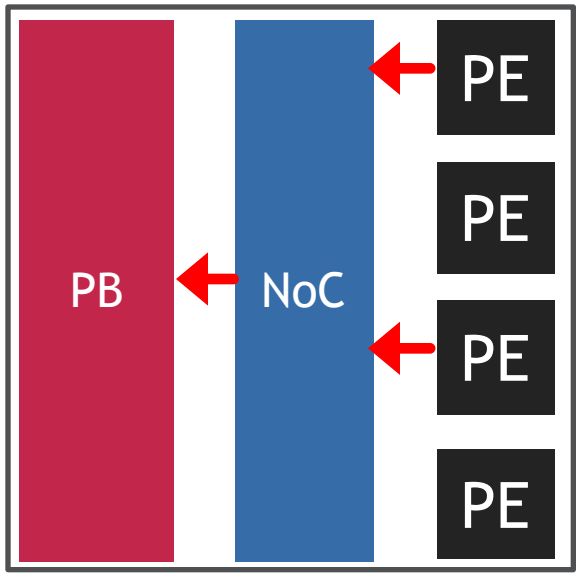
***How to enable flexible grouping?*** **Reconfigurable Interconnects!**

# Traffic Patterns in DNN Accelerators*



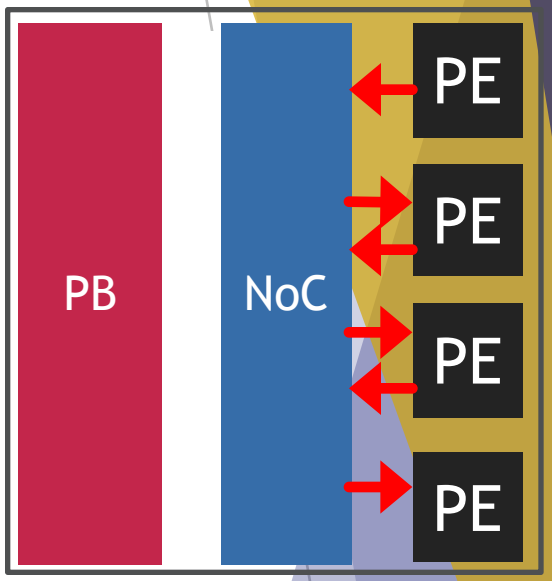

**Distribution**

**e.g.** input and weight distribution to PEs




**Reduction**

**e.g.** partial sum and output reduction




**Local Forwarding**

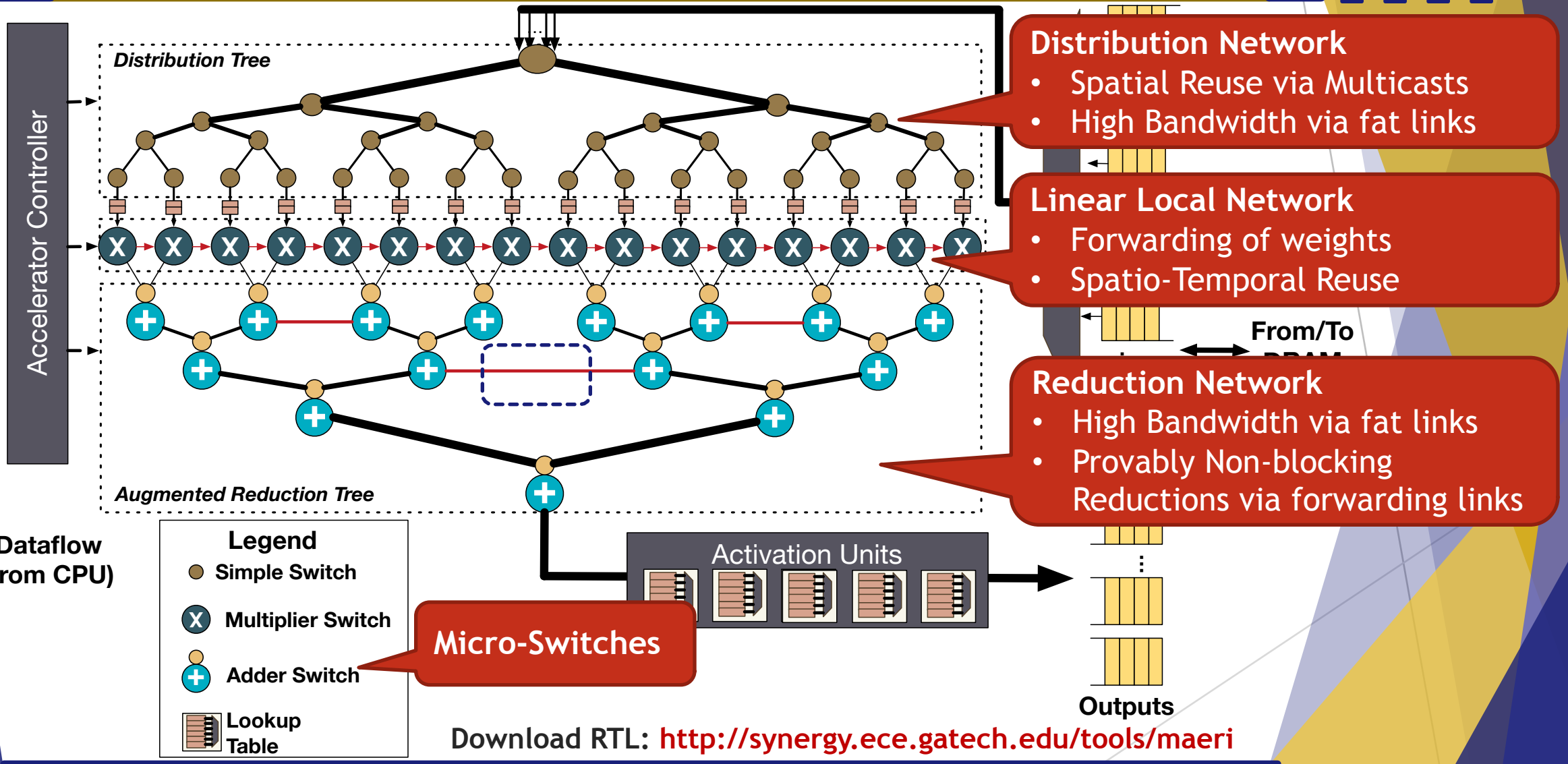
**e.g.** input/weight/partial-sum forwarding

* **PB:** Prefetch buffer (Global buffer)
* **NoC:** Network-on-Chip (Interconnection network)
* **PE:** Processing element (Compute units)



* H. Kwon et al., Rethinking NoCs for Spatial DNN Accelerators, NOCS 2017

# The MAERI Implementation




**Distribution Network**
- Spatial Reuse via Multicasts
- High Bandwidth via fat links

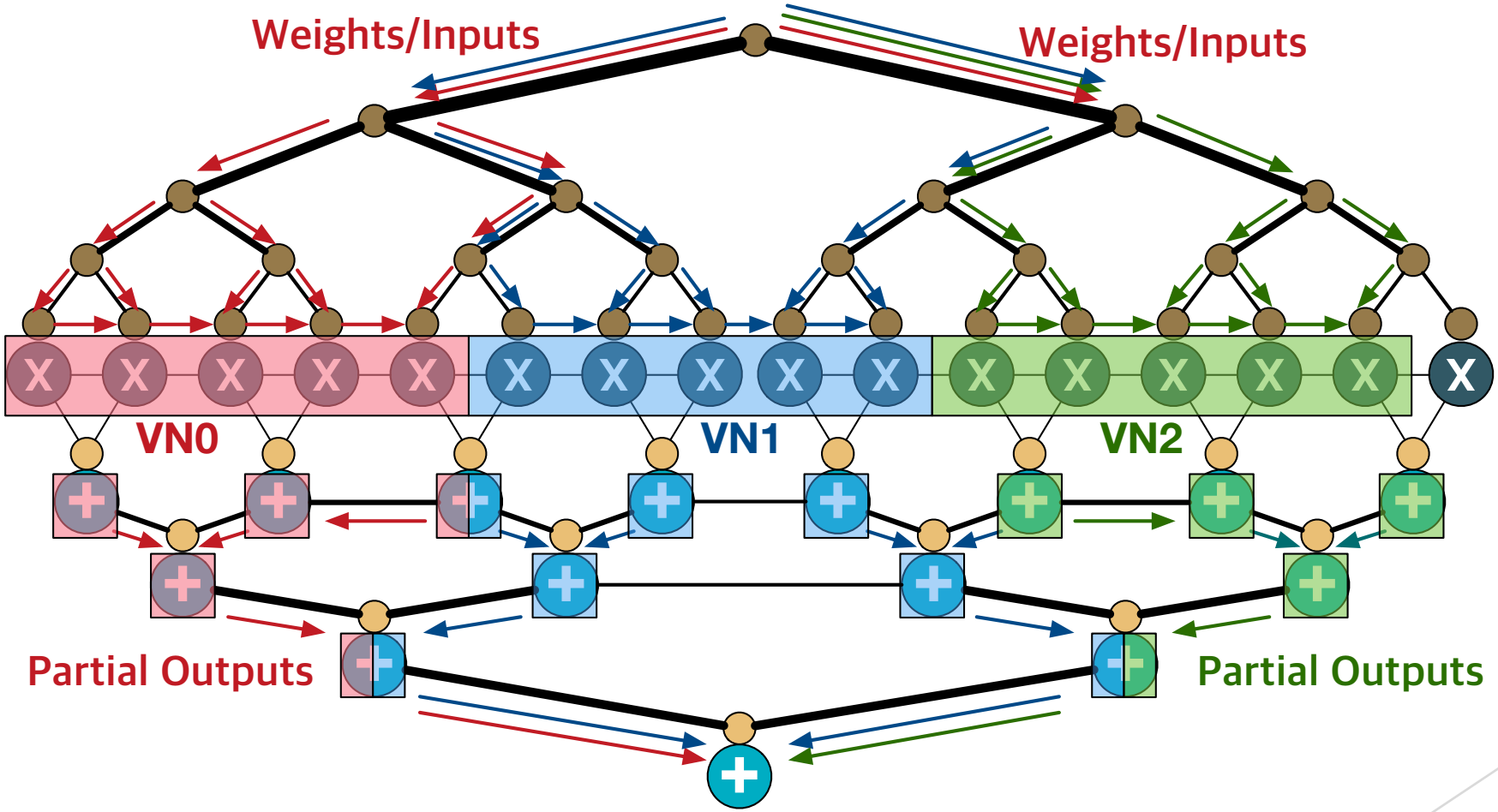
**Linear Local Network**
- Forwarding of weights
- Spatio-Temporal Reuse

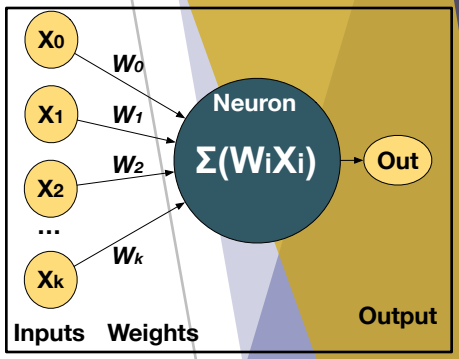
**Reduction Network**
- High Bandwidth via fat links
- Provably Non-blocking Reductions via forwarding links

Download RTL: http://synergy.ece.gatech.edu/tools/maeri

# Example Mapping – Dense CNN

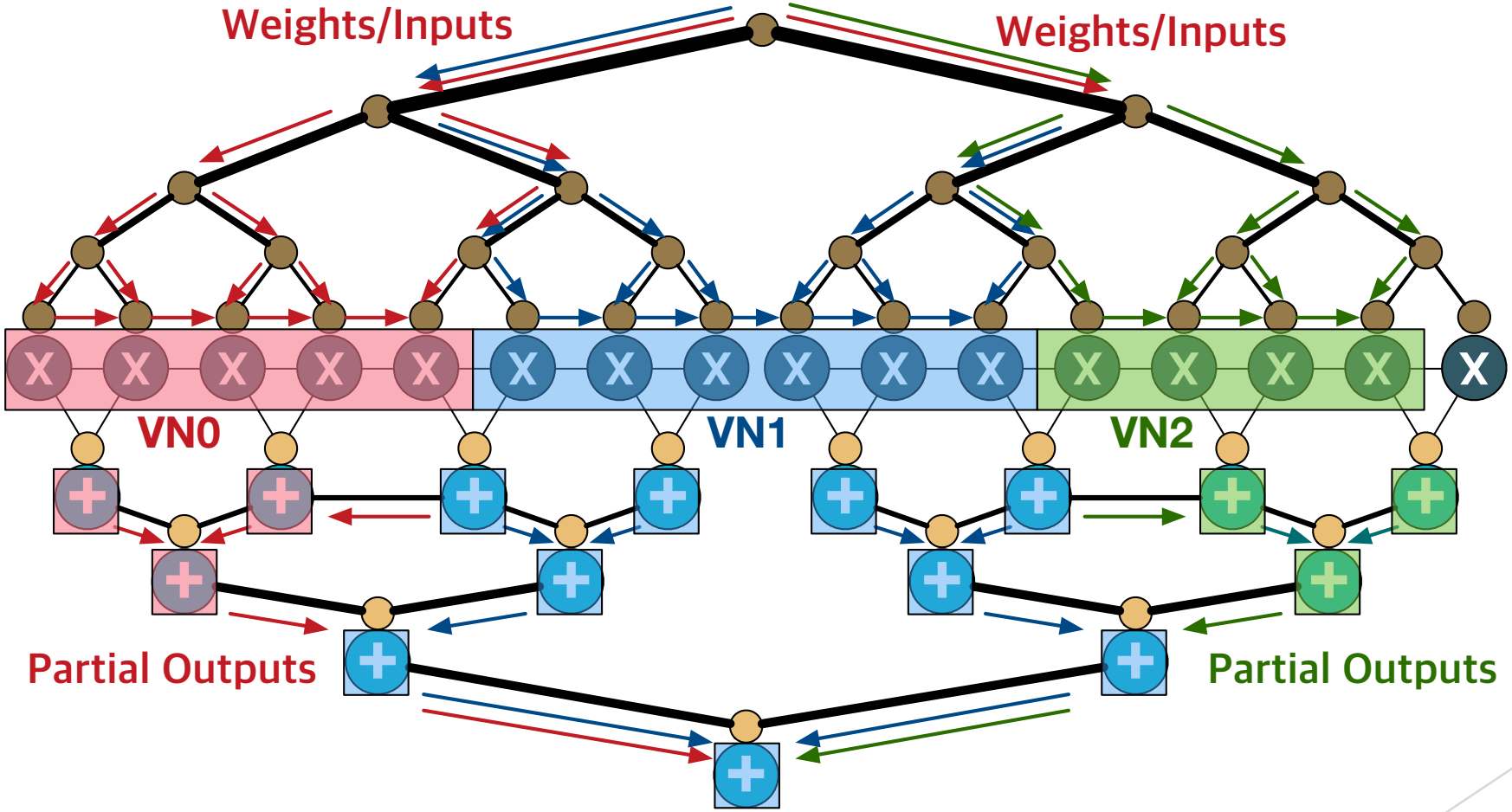
Our Key insight: Each dataflow translates into neurons of different sizes

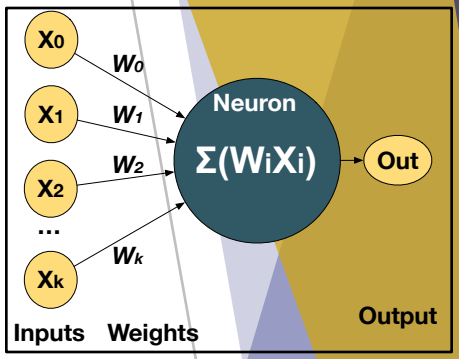


# Example Mapping – Sparse DNN

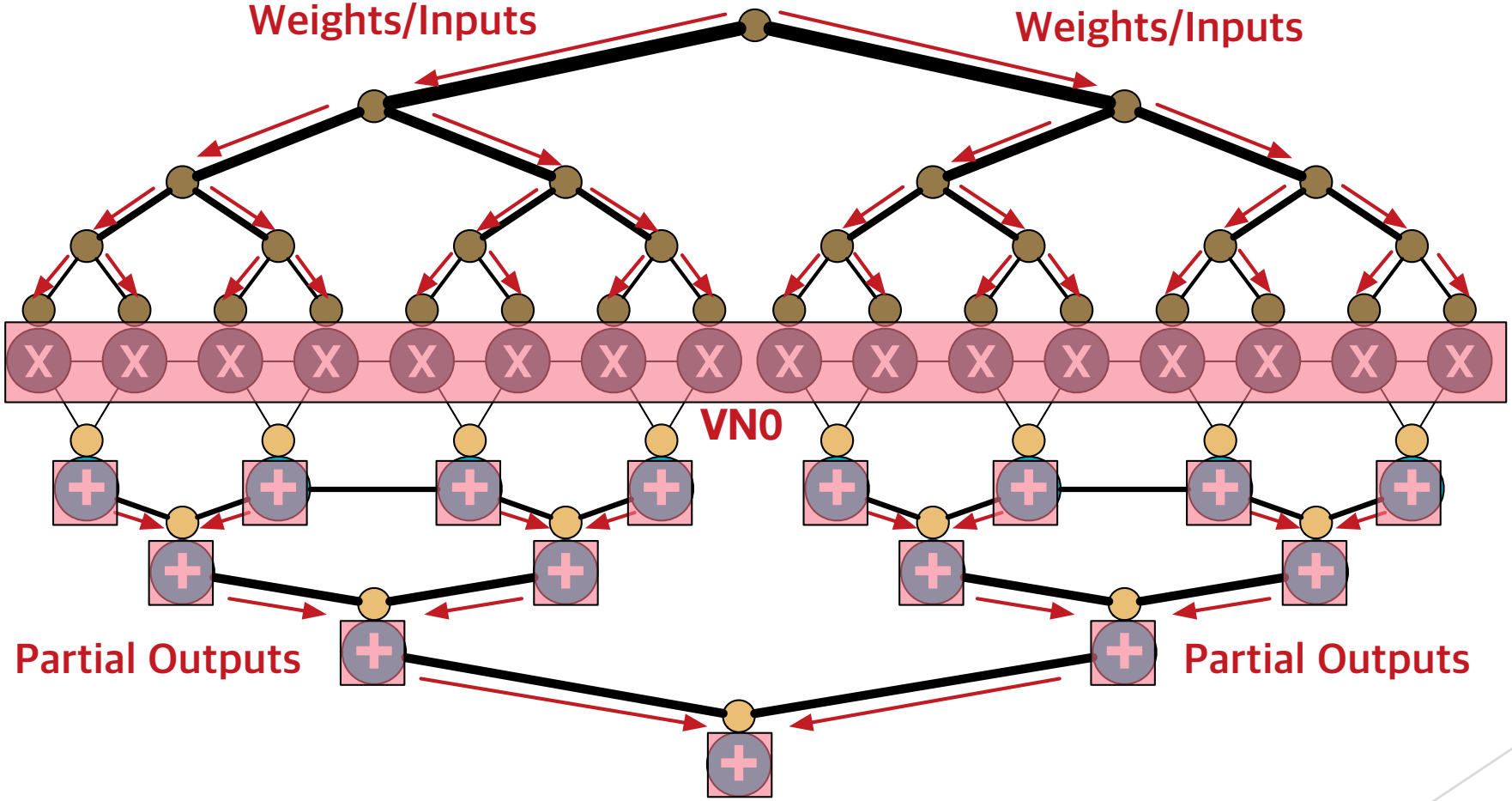
Our Key insight: Each dataflow translates into neurons of different sizes

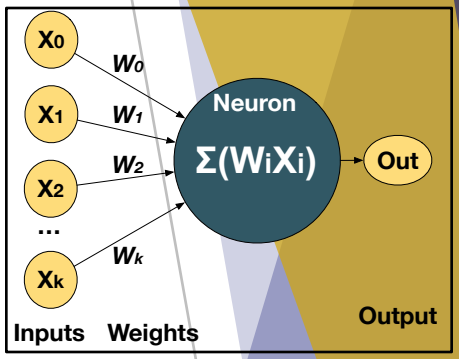


# Example Mapping – LSTM/FC

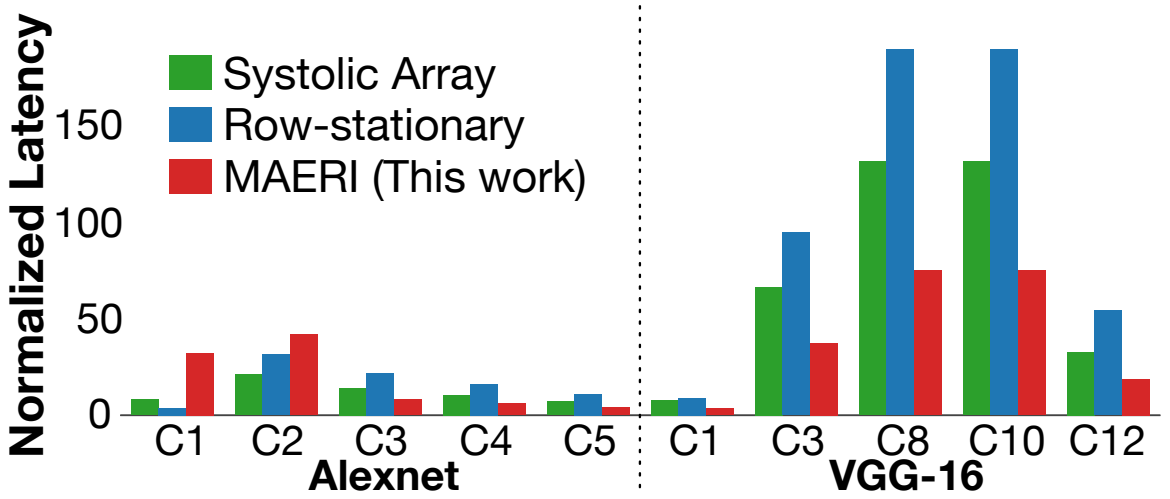
Our Key insight: Each dataflow translates into neurons of different sizes

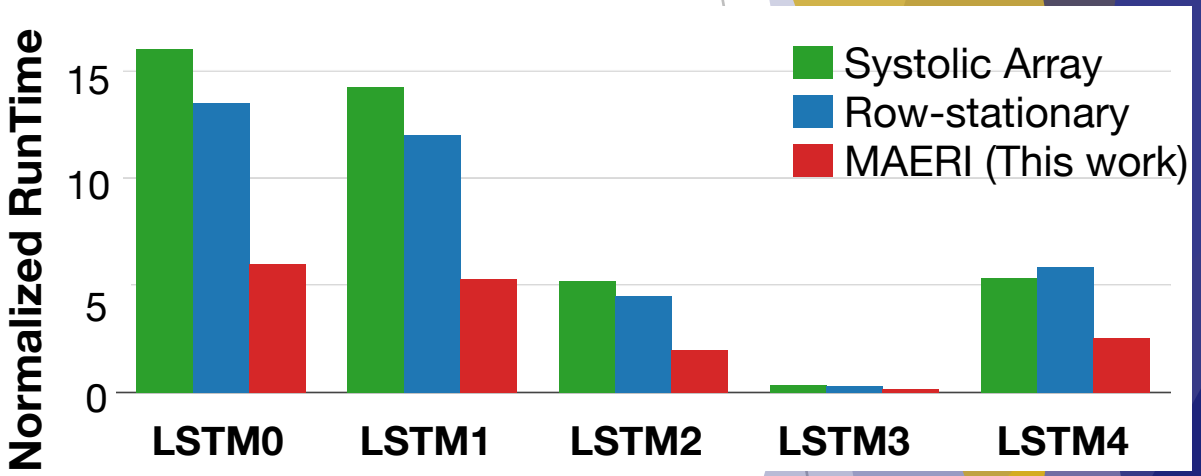


# Performance with Dense Workload

- **Total Latency (Runtime) for Convolution**




* Normalized to ideal case (100% utilization, Infinite bandwidth)

MAERI reduces CNN runtime upto 65%, 42% in avg.

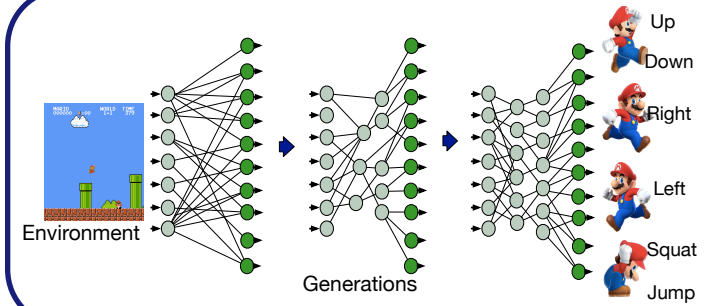
MAERI reduces LSTM runtime upto 63%, 57% in avg.

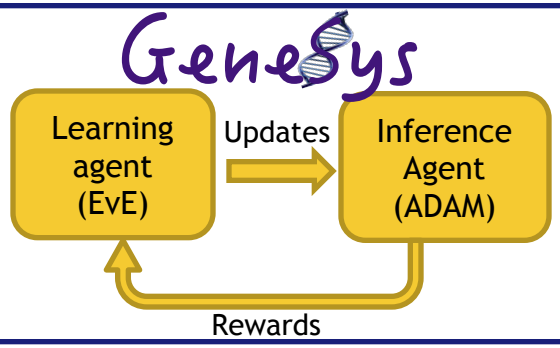
*LSTMs from Yonghui We, et. al., "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.", Arxiv Preprint, 2016*

# Takeaways

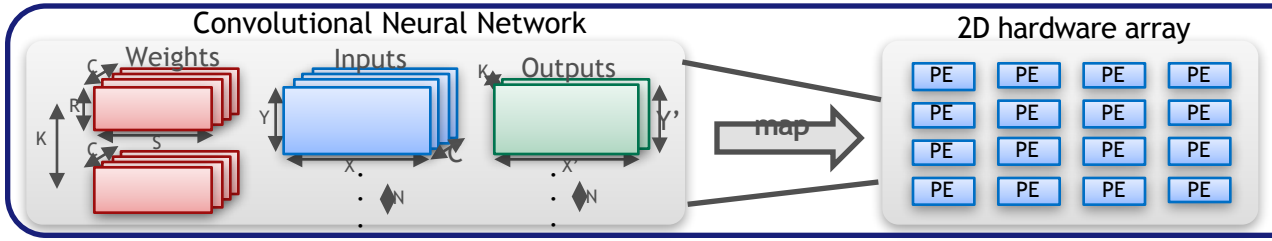
AI will be pervasive

Thank you!



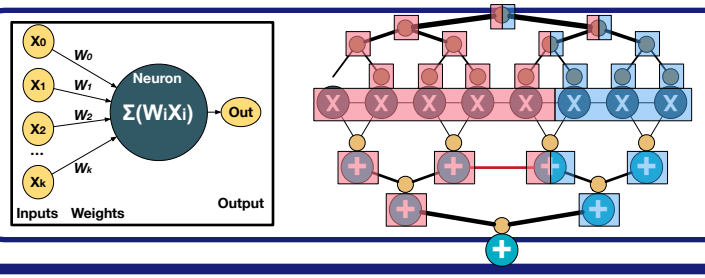




HW-SW Co-Design of NE Algorithms shows promise for continuous learning at the edge




MAESTRO

*Analytical Model for CONV Dataflow Analysis*




MAERI

DNN Accelerator with Configurable Interconnects can map Irregular Dataflows